Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Using Fuzzy Inference Systems for the Creation of Forex Market Predictive Models

AMAURY HERNANDEZ-ÁGUILA[1], MARIO GARCÍA-VALDEZ[1] AND JUAN-JULIÁN MERELO-GUERVÓS[2]

[1]National Technological Institute of Mexico, Calzada Del Tecnológico s/n, Fraccionamiento Tomas Aquino, Tijuana, BC 22414 Mexico (e-mail: amerhag,mario@tectijuana.edu.mx)

[2]University of Granada, Campus Aynadamar Daniel Saucedo Aranda s/n, Granada 18071, 80523 Spain (e-mail: jmerelo@geneura.ugr.es)

Corresponding author: Mario García-Valdez (e-mail: mario@tectijuana.edu.mx).

This paragraph of the first footnote will contain support information, including sponsor and financial support acknowledgment. For example, "This work was supported in part by the U.S. Department of Commerce under Grant BS123456."

ABSTRACT Financial market predictive models are tools that help us understand a market's behavior. Distinct methods for generating these predictive models are needed in order to obtain diverse insights about a market. This paper presents a method for creating forex market predictive models using multi-agent systems and fuzzy systems. Agents in our method represent traders performing buy and sell orders in a market. We use fuzzy systems to model the rules followed by traders performing trades in a live market to simulate its prices. Activation functions are used to restrict the agents' behavior, simulating the hesitancy of a trader when performing trades. Activation functions use the grades of membership obtained from an agent's fuzzy system, along with thresholds obtained from training data sets, to determine if that agent is specialized enough to handle a market's current conditions. We have performed experiments and compared against the work by Munkhdalai et al. [1], which demonstrates how capable our method is for predicting future market prices. Results demonstrate that our method is competitive compared to predictive models generated using deep learning, as well as models generated by random forest, AdaBoost, XGBoost, and support-vector machines. Furthermore, experiments where we compare our multi-agent systems with and without activation functions show that the use of activation functions to restrict the agents' behavior results in better predictive models.

INDEX TERMS Economic forecasting, fuzzy systems, multi-agent system, activation function, forex market.

I. INTRODUCTION

The prices of a financial market can be forecasted using different techniques, such as the analysis of raw price data or news involving the financial market of interest [2]. These approaches have the disadvantage of generating predictions entirely dependant on the trader's skills and knowledge about the market being predicted. A more robust approach when handling raw price data is the use of technical indicators [3], which preprocess a market's raw data to distill different aspects of it, such as a market's volatility or general direction. In the case of using news to draw predictions for a market, a more robust approach would be the use of sentiment analysis [4] [5], which can draw conclusions about the general sentiment of a financial market in order to know if prices will go down or up. An important drawback of these methods is that we are not obtaining explanations about why certain behaviors occurred; we are not generating a model about a market, we are simply distilling the data. Predictive models can be created using a variety of techniques, such as ARIMA [6] and hidden Markov models [7], or machine learning techniques, such as support-vector machines (SVM) [8] and neural networks [9]. A drawback of some of these methods is that they generate *black box* models, i.e. it is difficult for a user to understand how the model is processing its inputs. A technique that alleviates this problem is fuzzy logic, which can also be used to create predictive models [10] [11] [12].

The method presented in this paper uses a hybridization of fuzzy logic and multi-agent systems (MAS). The use of fuzzy logic enables the method to generate easily interpretable models to the user, especially if we use Mamdani fuzzy sets [13] to create the membership functions of a fuzzy system, as in the work of Abdulgader and Kaur [14]. In particular, we propose the use of intuitionistic fuzzy logic (IFL) [15] [16], as IFL adds another layer of interpretability for the fuzzy systems through the concept of indeterminacy. Although there are alternatives to IFL, such as type-2 fuzzy systems, the defuzzification of IFL systems can be faster than the aforementioned systems [17], while also providing comparable interpretability.

Fuzzy logic has proved to be useful in many fields, and one can find many works involving financial market prediction using fuzzy systems, such as the works by Tsai et al. [18] and Zeng et al. [19], where fuzzy time-series are used to perform the predictions; as in the works by Rajab and Sharma [20] and Vlasenko et al. [21], where a neuro-fuzzy approach is taken; or as in the works by Yue et al. [22], Witayakiattilerd [23] and Mansour et al. [24], where fuzzy logic is used to perform a portfolio selection of financial markets.

The present work uses fuzzy systems in combination with MAS to create a market model, where the agents in the MAS represent traders, and the trades performed by these traders are used to simulate the real prices of a market. MAS have been demonstrated to be an effective approach to simulate very complex systems, such as those represented by financial markets [25] [26]–[29]. Additionally, the agents in the MAS can be examined to understand how individual traders are interacting in a financial market [30] [20], which is a feature that we leverage in our method, by using fuzzy systems to construct the agents' rules, resulting in interpretable inference systems. Furthermore, MAS architectures can be executed in a distributed manner [31] [32] [32], enabling faster performance due to the rules of diferent agents being evaluated in parallel. The current implementation of our method does not follow such architecture, but adapting our method to a distributed evaluation of the agents' rules is considered as a future work, as is discussed in Section VIII.

The reader will find an in-depth explanation of our method in Section III, and explanations for the concepts required to understand the method can be found in Section II. A description of an implementation of our method can be found in Section IV. In order to evaluate the perforance of our method, experiments were performed and are described in Section V. The results of the experiments are presented in Section VI, and a discussion of these results can be found in Section VII. As a prelude, we find that: i) our method describes a novel architecture that mixes MAS and fuzzy systems for the creation of predictive models; ii) the activation functions found in the agents' fuzzy systems help to decrease the error between predicted and real market prices; iii) the results from the experiments indicate us that the models generated by our method are competitive against models generated by deep learning, random forest, AdaBoost, XGBoost and SVM; iv) our method describes a simple and effective method for the tuning of parameters for MAS. Finally, Section VIII discusses some directions that our presented method can take in the future to better demonstrate its capabilities.

II. PRELIMINARIES

This Section describes the concepts that the reader needs to be familiar with in order to better understand the proposed method in Section III.

A. FUZZY SETS

A traditional set is a collection of items that share a common characteristic. This characteristic serves as a membership, because all the items in a universe either have that characteristic—and then the item is part of the set—or it does not have it—and then the item is not part of the set. Traditional sets can be extended to fuzzy sets, as explained by Zadeh [33]. Fuzzy sets are then a generalization of traditional sets, i.e. any traditional set can be represented as a fuzzy set. The difference between these two type of sets lies in the concept of membership: memberships are not only used to represent binary outcomes, i.e. *true* or *false*, but now a possibly infinite number of outcomes. An item can now be partially a member of a set, and the only way an item is not part of such set is if its membership is totally *false*. In order to represent this grade of membership one can use real numbers. Thus, one can say, for example, that an item is *0.7 green*, *0.5 blue* and *0.0 red*. These values can represent an adverb and an adjective, such as "very green," "somewhat blue" and "not red at all." This is especially useful when designing fuzzy systems (see Subsection II-B).

B. FUZZY SYSTEMS

In traditional logic one can generate logical inferences, such as *if it's raining, then there are clouds in the sky*. In a similar fashion, we can use fuzzy sets to represent the antecedents and consequents in a logical inference process [34]. For example, one can extend the previous example to: *if it's raining a lot, then there are many clouds in the sky*.

There is a number of ways in which one can construct a fuzzy inference system, where one or more inputs or antecedents can be used to generate one or more outputs or consequents. Arguably, the two most popular types of fuzzy inference systems are the ones proposed by Mamdani and Assilian [13], and Takagi and Sugeno [35]. These systems use a series of fuzzy sets to represent the relationship between an input and its grade of membership to a set. These sets usually represent adjectives that describe the inputs, and are also considered to be the antecedents in the fuzzy inference system. For example, an input of 0.8 can represent a "very high" value. After obtaining these grades of membership, one can use these values to "fire" or "activate" the consequents. In the case of a Mamdani system, the consequents are represented as fuzzy sets, just like the antecedents. In contrast, in a Sugeno system, consequents are represented by mathematical functions. A set of rules is used to determine the relationship between the antecedents and the consequents, for example: *if food quality is high then tip*

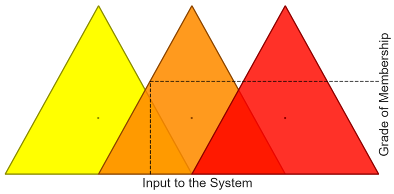

FIGURE 1. Example of antecedents in a Mamdani fuzzy system.

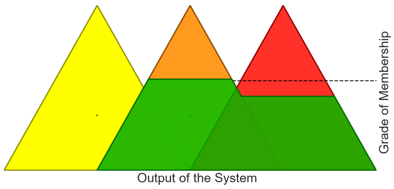

FIGURE 2. Example of consequents in a Mamdani fuzzy system.

is high. The aforementioned rule is creating a relationship between the fuzzy set that represents "high food quality" in the antecedents, and the fuzzy set that represents "high tip" in the consequents. Further continuing with the example, if "food quality" is represented by a value of 0.8, the rule that creates the relationship between "food quality" and "tip" could determine a "tip" of 0.8 too, depending on what membership function and what parameters are decided to be used to represent each.

We have explained how a relationship between antecedents and consequents can be constructed in a fuzzy inference system. Nevertheless, the most interesting problem arises when a problem involves several fuzzy sets to represent different adjectives for single antecedents or consequents. In these cases, depending on the fuzzy rules, a number of consequents can be fired according to the inputs to the system. As seen in Figure 1, the input—represented by the dotted vertical black line—is associated with three fuzzy triangular sets or antecedents, where it "activates" two of them. According to a set of fuzzy rules, the inputs then fire a set of triangular fuzzy sets that represent the consequents, as seen in Figure 2.

The fuzzy sets that represent the consequents are cut, and new shapes are obtained using those cuts, as represented by the green shapes in Figure 2. These shapes are aggregated and result in the output of the fuzzy inference system, and this result can then be defuzzified using different methods, such as obtaining the centroid of the shape. In this example, a Mamdani fuzzy inference system is considered; in the case of a Sugeno system, for example, the antecedents would be represented by arbitrary mathematical functions, instead of membership functions representing shapes such as the triangles in the example presented above.

C. INTUITIONISTIC FUZZY SETS

In contrast to the traditional fuzzy sets discussed in Subsection II-A, intuitionistic fuzzy sets consider a grade of non-membership in addition to a grade of membership associated to an element in the fuzzy set [15], as expressed by 1.

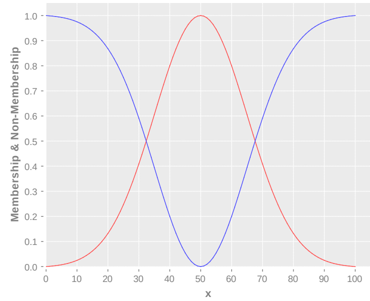

FIGURE 3. Traditional fuzzy set represented as an intuitionistic fuzzy set.

$$A^* = \{\langle x, \mu_A(x), \nu_A(x)\rangle | x \in E\} \quad (1)$$

For every one of the elements contained in an intuitionistic fuzzy set, 2 must hold true.

$$0 \leq \mu_A(x) + \nu_A(x) \leq 1 \quad (2)$$

Intuitionistic fuzzy sets are an extension to traditional fuzzy sets, as any traditional fuzzy set can be expressed as a particular case of an intuitionistic fuzzy set, as in 3.

$$\{\langle x, \mu_A(x), 1 - \mu_A(x)\rangle | x \in E\} \quad (3)$$

If the sum of the membership $\mu_A(x)$ and non-membership $\nu_A(x)$ of an element is less than 1, the concept of indeterminacy or hesitancy arises [15], which is described by 4. Indeterminacy is used to represent doubt in the grade of membership of an element in an intuitionistic fuzzy set and is described by 4.

$$\pi_A(x) = 1 - \mu_A(x) - \nu_A(x) \quad (4)$$

Traditional fuzzy sets can be extended to increase their capabilities of representing uncertainty by introducing the concept of footprint of uncertainty [36] [37]. A footprint of uncertainty is achieved by extending the membership function, where each value transforms from a crisp value into a fuzzy set. Indeterminacy serves a different purpose than that of footprint of uncertainty. Instead of extending the uncertainty provided by traditional fuzzy sets, indeterminacy helps to model doubt. For example, if traditional fuzzy sets can model the following sentence: "the object is very hot", indeterminacy can model "it is unsure that the object is very hot".

D. INTUITIONISTIC FUZZY SYSTEMS

Intuitionistic fuzzy sets, like traditional fuzzy sets, can be used to create inference systems. Antecedents and consequents in the system can be handled by intuitionistic fuzzy sets, as in a Mamdani system [13]; alternatively, they can be used solely for the antecedents, with mathematical functions used for the consequents as in a Sugeno system [35].

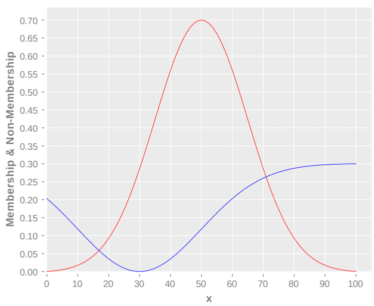

FIGURE 4. Intuitionistic fuzzy set with membership and non-membership functions with different means and standard deviations.

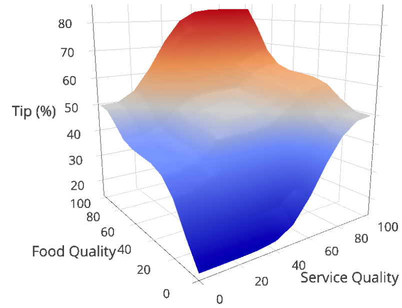

FIGURE 5. Output surface for the tipping problem using a traditional fuzzy system.

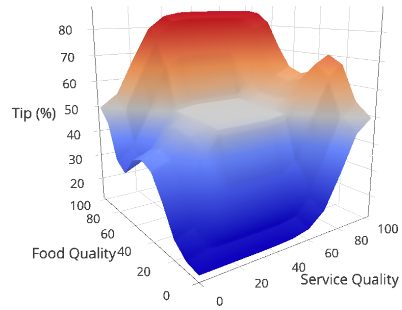

FIGURE 6. Output surface for the tipping problem using an intuitionistic fuzzy system.

Various approaches have been taken in the past by different authors on how to build intuitionistic fuzzy systems. The authors of the present paper have worked in a certain way to achieve this type of system, and this method is described in [17] and [38]. The method described in the aforementioned works is presented in this Subsection for the reader as a reference implementation.

As it is explained in [17], in an IFIS, in order for an antecedent to fire a consequent according to a set of fuzzy rules, the final grade of membership of an element has to be expressed in terms of its grade of membership and its grade of non-membership. The resulting grade of membership of an element belonging to A is represented by $i\mu_A(x)$, and is defined in (5).

$$i\mu_A(x) = (\nu_A(x) + \mu_A(x))\mu_A(x) \tag{5}$$

To perform an alpha-cut in a consequent, one has to separate it in two stages: i) first, perform a traditional alpha-cut in the membership function following equation (6), and then ii) perform an alpha-cut in the non-membership function following equation (7).

$$\alpha(\mu(x), \mu_\alpha) = \begin{cases} \mu(x), & \text{if } \mu(x) \leq \mu_{alpha} \\ \mu_\alpha, & \text{otherwise} \end{cases} \tag{6}$$

$$\alpha_{NMF}(\nu(x), \mu_\alpha) = \begin{cases} \nu(x), & \text{if } \nu(x) \geq \nu(\mu_{alpha}) \\ \nu(\mu_{alpha}), & \text{otherwise} \end{cases} \tag{7}$$

The aggregation of the fired consequents is performed by applying (8) on the alpha-cuts.

$$\begin{aligned} A \cup B = \{ & \langle x, max(\mu_A(x), \mu_B(x)), \\ & min(\nu_A(x), \nu_B(x))\rangle | x \in E\} \end{aligned} \tag{8}$$

The final modification to the traditional inference process in a FIS is made to the center of area procedure. The equation to calculate the center of area of a traditional fuzzy set is (9). In order to implement a center of area for an intuitionistic fuzzy set, one has to incorporate the concept of $i\mu(x)$, giving as a result (10), and its simplification form (11).

$$A_{CoA} = \frac{\sum_{i=1}^{N} \mu(x_i)x_i}{\sum_{i=1}^{N} \mu(x_i)} \tag{9}$$

$$A_{iCoA} = \frac{\sum_{i=1}^{N} (\mu(x_i) + \nu(x_i))\mu(x_i)x_i}{\sum_{i=1}^{N} (\mu(x_i) + \nu(x_i))\mu(x_i)} \tag{10}$$

$$A_{iCoA} = \frac{\sum_{i=1}^{N} i\mu_A(x)x_i}{\sum_{i=1}^{N} i\mu_A(x)} \tag{11}$$

E. MULTI-AGENT SYSTEMS AND AGENT-BASED MODELS

In our method, fuzzy systems are used to model a trader's knowledge and how they take trades according to a market's current conditions, and MAS are used to model the collective traders of a market and their respective influence on the price of it. MAS create agent-based models, which represent a problem that people can analyze and infer new knowledge from it [26].

Agents can be seen themselves as programs that interact with their environment, which may include other agents. MAS are composed of different, autonomous entities with "beliefs" and "rules". Beliefs are used by agents to arrive to an interpretation of their environment, and rules are used to arrive to actions to be performed by the agent towards their environment. Agents in the system are constantly assesing their environment to determine what actions to take according to their beliefs and rules. MAS have the objective of solving a practical problem, unlike agent-based models which are more focused on simulating it. Both use the same tools, and only intent is different.

The proposed method involves the use of a MAS which acts in a decentralized fashion. However, some of its mechanisms are centralized: the outputs of the agents are averaged, and agents receive their inputs from the same source.

III. PROPOSED METHOD

The following subsections describe the proposed method in detail. Subsection III-A describes the structure followed by each of the agents that shape a predictive model, while Subsection II-E describes how the agents in the predictive model are coordinated and how they interact with their environment. Finally, Subsection III-C describes an iterated local search (ILS) method that finds a combination of agents that is suitable to be used as a predictive model for particular data sets.

A. AGENT ARCHITECTURE

Agent rules are defined by Mamdani intuitionistic fuzzy inference systems. Fuzzy systems are convenient, as they can be interpreted, as opposed to, for example, neural networks, where the weights associated to the neurons and the connections among themselves become obscure to interpretation. Furthermore, the use of intuitionistic fuzzy systems provides an additional layer for interpretation: indeterminacy.

Indeterminacy arises as a consequent of the inclusion of non-membership to a fuzzy system (see Subsection II-C). This concept allows the designer of a fuzzy system to model doubt or hesitancy in a data set. In the proposed method, indeterminacy is obtained in a heuristical manner, as part of the optimization algorithm that searches for a combination of agents that are used to create a predictive model.

The membership functions in the fuzzy systems are always defined as Gaussian functions, although in future experiments this design choice can change, as other types of membership functions could provide benefits over Gaussian membership functions, such as better interpretability for particular problems being modelled, and improvements in computational cost. Gaussian membership functions were chosen because of their ability to model knowledge in a smoother way than their alternatives, such as triangular or trapezoidal membership functions. Although other membership functions can be better suited for certain situations, the proposed method is currently designed for the creation of predictive models for arbitrary data sets, where solutions are found using ILS.

In the proposed method, the mean of each Gaussian membership function is equal to a random data point from the training data set, while the spread of the Guassian membership functions that represent a fuzzy rule will be equal to the standard deviation of the aforementioned randomly chosen data points. At least two Gaussian membership functions are used to describe each agents' rule antecedents and consequents for, as obtaining the standard deviation of only one data point would be equal to 0, which would generate Gaussian membership functions with spreads equal to 0. In the case of the antecedents, the means are equal to data points from the training data set that represent inputs, while outputs in the training data set are used as the means for the membership functions that form the consequents. This is expressed by 12, where $\bar{x}$ represents the mean of sample inputs from the training data set, s represents their standard deviation and $\mu(x)$ represents a Gaussian membership function.

$$\mu(x) = e^{-\dfrac{(x-\bar{x})^2}{2s}} \tag{12}$$

Data points are used as the mean of each membership function to guarantee an agent's ability to react to at least those input values, and thus every agent will respond to at least one data point, which is a common practice in certain methods, such as rule-based classifiers [39] [40]. On the other hand, using the standard deviation of those chosen data points as the spread of the Gaussian membership functions guarantees that uncertainties associated to each membership function will affect—in terms of fuzzy inferences—at least one of the other membership functions. If none of the membership functions were affecting, in any way, the rest of the membership functions, the use of a fuzzy system to represent the agent rules would be meaningless.

The domain of each membership function is not fixed as it is determined by the chosen data points and their standard deviation. The domain of a membership function is defined by the set given by 13, where $\bar{x}$ represents the mean of sample inputs from the training data set and s represents their standard deviation. As a consequent of the previous definition, the domain of either the antecedents or the consequents in a fuzzy system is defined by the set given by 14, where $\bar{x}_{min}$ and $\bar{x}_{max}$ represent the minimum and maximum means—from the set of means obtained from the training data set–used to define the membership functions, respectively, and s_{min} and s_{max} represent the minimum and maximum standard deviations that are associated to $\bar{x}_{min}$ and $\bar{x}_{max}$, respectively.

$$\{\, x \mid \bar{x} - s \leq x \leq \bar{x} + s \,\} \tag{13}$$

$$\{\, x \mid \bar{x}_{min} - s_{min} \leq x \leq \bar{x}_{max} + s_{max} \,\} \tag{14}$$

As the agents' rules are represented by intuitionistic fuzzy systems, the core of the membership functions is not necessarily equal to 1—or it does not exist, if one considers the core of a membership function to be restricted to a value of 1—, as it is the case in traditional fuzzy systems [41]. The value of the greatest grade of membership in a membership function is determined heuristically using an optimization algorithm that is explained in Subsection III-C.

Indeterminacy is used to "fuzzify" activation thresholds associated to inputs to the agent, which determine if the agent should respond to that input or not. Two agents with the same

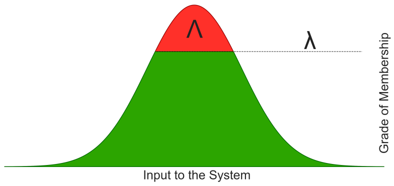

FIGURE 7. Depiction of an activation threshold λ and a set Λ in a membership function.

fuzzy rules, membership functions and activation thresholds will respond differently to the same inputs if they have different non-membership functions. This way, the system uses uncertainty—coming from membership functions—and indeterminacy—coming from non-membership functions—to model the membership of an input and an activation function, respectively. Considering the agent system architecture proposed by Shoham [42], in the proposed method an agent's fuzzy system represents an agent's rules, while its activation functions represent an agent's beliefs.

Inputs to an agent's fuzzy system are associated to grades of membership—as is usual in fuzzy inference systems—and this grade of membership can either belong or not to a set Λ, which is defined by 15, where $\mu(x)$ represents the grade of membership associated to an input x and λ represents an activation threshold. An example of a set Λ can be seen in Figure 7. If a grade of membership can be found in a set Λ, then the agent will consider that input to be used to fire a consequent in its agent rules. It is noteworthy that different activation thresholds λ and non-membership functions can be associated to each of the membership functions present in the antecedents of an agent's fuzzy system.

$$\mathbf{\Lambda} = \{\, \mu(x) \mid \mu(x) \geq \lambda \,\} \tag{15}$$

An agent's actions can be greatly limited by its activation functions, as having a single input associated to a grade of membership which does not belong to the activation interval Λ is sufficient for an agent to take no action. This behavior allows to precisely control the magnitude of specialization of each agent in a predictive model, as the designer of the model—such as an optimization algorithm—can assign activation thresholds and non-membership functions that restrict an agent to be activated to only a handful of inputs from a training data set. Furthermore, activation functions work as a coordination mechanism for the agents in a predictive model, as they prevent certain agents from taking action in situations where they would perform sub-optimally and others would perform optimally.

B. MULTI-AGENT SYSTEM

Predictive models that follow the proposed method are formed by a set of one or more agents constructed using the architecture described in Subsection III-A, resulting in a MAS.

Agents must work together in order to simulate a financial market. One way of achieving this is to obtain the output of every agent in the predictive model and to use an aggregation process to unify them into one single output. Instead, the proposed method uses activation functions to restrict what agent outputs are used to respond to a set of inputs. This is similar to what happens in a real market: the aggregation of all the trades, from every trader, result in the current price of that market. Furthermore, although traders could decide how to trade a market at any given point, traders sometimes restrict themselves from trading because they consider a market's current condition to not be ideal.

The restriction imposed by the activation functions ensures that every agent is specialized at different subsets of the training data set. During the creation of the agent, the agent is tested with each of the input data points from the training data set to compile a list of activation levels—i.e., the grades of membership associated to each of the inputs, considering the membership functions in the antecedents of the agent's rules. Once the list of activation levels is compiled, the list is sorted by using the sum of the activation levels. After compiling the sorted list of activation levels, the proposed method chooses one of them, from highest to lowest activation, according to a *depth* parameter. This process is depicted in Algorithm 1.

Algorithm 1 Selection of activation threshold

```
procedure ACTIVATION-THRESHOLD(A, I);
    ant_A ← extract_antecedents(A)
    c ← 0
    for each inp ∈ I do
        acts ← activations(ant_A, inp)
        sum[c] ← Σ_{i=1}^{n} acts_i
        c ← c + 1
    end for
    return sum(DEPTH)
end procedure
```

The chosen activation level serves as an activation threshold, as any set of inputs that do not activate all the membership functions according to the activation level will fail to activate the agent to take an action. As a consequent—during the training stage of the method—any agent will be activated to a number of inputs equal to the value of the *depth* parameter shown in Algorithm 1.

An implication of the aforementioned process is that activation functions also create a restriction for the actions of the agents in a predictive model. This restriction works as a coordination mechanism to ensure agent participation in the prediction process never reaches one hundred percent, as certain inputs will not trigger a response from any of the agents. In other words, the predictive model only outputs a response if, and only if, the agents have learned a pattern with a strong resemblance to certain inputs.

Although the activation functions cause the agents to be specialized at responding to a number of inputs, an activation of multiple agents is possible. In this case, the outputs of the agents—positive or negative numbers, which represent buy or sell orders, respectively—are averaged. We have the hypothesis that activation functions improve the performance of the models for the forecasting of forex markets, as agents only respond to those inputs where grades of membership in the fuzzy system's antecedents are the highest or, in other words, agents are restricted to respond to those inputs that are similar to those used to create the agents' membership functions.

The process of agent specialization using activation functions is well suited for the creation of predictive models where it is not desired to obtain a response for any arbitrary set of inputs. In the case of financial market forecasting, this is translated to a recommandation of not trading a particular market, i.e. an unkown pattern is arising and the trader following the recommendations from the predictive model should be wary.

However, it must be noted that the proposed method can be extended to the creation of predictive models that always yield a response. This can be achieved by selecting the agent that is closer to being activated.

Finally, we believe that the activation functions found with the proposed method could be used for other methods, particularly neural network-based architectures.

C. PARAMETER TUNING USING ITERATED LOCAL SEARCH

The sets of inputs and outputs that are used as the means of the membership functions in the agent rules are determined randomly—as we explained in Subsection III-A. In order to obtain a predictive model, ILS is used to find combinations of agents that generate a suitable simulation of a financial market.

The proposed method uses a basic search algorithm that adds and/or removes agents from a list of agents, where these agents work together to create a simulation of the market. The list of agents begins at iteration 0 with a single, random agent, and in the following iterations it is randomly decided to either add new agents or remove them from the list. The modifications are committed only if the addition or removal operation improves the performance of the predictive model, and the algorithm finishes after a number of iterations has been reached. The performance improvement is measured with a loss function, which is used to determine if a predictive model is better than its predecessor. This algorithm is depicted in detail in Algorithm 2.

As is mentioned in Subsection II-E, the resulting predictive model does not necessarily respond to every set of inputs. This behavior is accentuated when the model is tested against a data set that is different than the one used for the training stage of the method—i.e., a testing data set—, as inputs from this data set could not resemble at all any of those present in the training data set. We have the hypothesis that this behavior helps the generated models achieve better performances.

Algorithm 2 Iterated local search used to find a solution in the proposed method

procedure ILS(S);
    $push(make\_agent(), S)$
    $c \leftarrow 0$
    while $c < ITERATIONS$ **do**
        if $random() < 0.5$ **then**
            $candidate \leftarrow append(make\_agent(), S)$
        else
            $candidate \leftarrow remove\_random\_agent(S)$
        end if
        if $eval(candidate) < eval(S)$ **then**
            $S \leftarrow candidate$
        end if
    end while
    return S
end procedure

IV. IMPLEMENTATION

The agents in the prediction models are represented as objects with the following properties: agent's rules, which stores an intuitionistic fuzzy system; activations, which stores the antecedents of the fuzzy system; and activation thresholds, which stores the levels that need to be surpassed by a set of inputs in order to activate the agent.

Indeterminacy in an agent's fuzzy system is determined randomly and optimal values are found with the ILS method described in Subsection III-C. In order to obtain intuitionistic fuzzy sets where 2 holds true, our implementation generates membership functions where the greatest possible grade of membership is M and non-membership functions where the greatest possible grade of non-membership is $1 - M$. The mean and spread of the Gaussian membership functions are equal to the mean and spread of the Gaussian non-membership functions in all cases.

The agents that are randomly generated for the ILS need to pass a test before they can be considered a candidate to be part of a predictive model. Considering a training data set, an agent's activation functions are evaluated against the inputs set obtained from that training data set in order to generate a set of activations, where each activation is the equivalent of calculating the grade of membership associated to a set of inputs. After performing this step, all the activations are summed to obtain a score S that represents the intensity to which that agent reacts to that set of inputs. This score is defined by 16, where N represents the number of inputs, and $\mu(x_i)$ represents each of the grades of membership associated to each input. The scores associated to each set of inputs are ordered in a descending manner, so the first elements are the scores that represent those sets of inputs that would fire an agent's consequents the most. The ordered scores list is also used to obtain the outputs that are associated to these ordered scores, so we can know what are the outputs that the agent *should* evaluate to given those particular sets of inputs. The resulting set of outputs are used to determine if the agent is a suitable candidate; if most of the outputs associated to the highest scores have the same direction (negative or positive), then it is a suitable candidate. This mechanism ensures that the chosen agents are specialized in similar inputs that yield similar outputs. For our implementation, the highest scores have to be associated to outputs that share the same direction at least 66% of the time. This value was chosen empirically after performing some preliminary tests, where different values greater than 50%—as we want the majority of the outputs to share the same direction—were used. However, it is not statistically proven that this value will yield better results than other values.

$$S = \sum_{i=1}^{N} \mu(x_i) \tag{16}$$

The optimization process for the predictive models loads configuration files as its first step, depending on the market that we want to use to obtain training and testing data sets. These configuration files set parameters for the proposed method, such as the train-test ratio and number of inputs, outputs and number of rules for the agents' fuzzy systems.

For our implementation of the proposed method we chose Common Lisp as our language, so we could use a software library for the creation of intuitionistic fuzzy systems that we implemented in the past [17] [38]. All the populations are compressed and stored in a PostgreSQL database, which enables us to resume the optimization of a model at any time. Storing the populations in a database also enables us to use populations of agents to be tested in other data sets, as well as to extract agents from certain populations to be used in other prediction models. The source code of our implementation can be found in the git repositories at this link: https://bitbucket.org/overmind-group/workspace/projects/OT.

The evaluation of the agents proved to be a computationally expensive task, as the system needs to evaluate hundreds of fuzzy systems per iteration in the optimization process. For this reason, we implemented a caching system where the output of an agent is stored in memory using a technique called *memoization* [43] in a functional programming context. After *memoizing* an agent, if that agent is required to be evaluated with exactly the same inputs as the ones used during the *memoization* process, then we can assume that the output will be the same, and thus we can simply query for the output stored in memory. The caching system prevents our implementation from re-evaluating hundreds of fuzzy systems in the optimization process.

V. EXPERIMENTS

In order to determine the efficacy of the proposed method, we designed experiments that involved data sets and performance metrics used in the work by Munkhdalai et al. [1]. In [1], daily prices for the following Forex markets are used: EUR/USD, USD/JPY, USD/CHF, GBP/USD, USD/CAD and AUD/USD. The last year of prices are used as their data set, although they do not provide the exact starting and ending dates. Their datasets are partitioned into three parts: training, validation and test sets, where the training set corresponds to 80% of the data set, the validation set corresponds to 10% of the data set, and the test set correspond to 10% of the data set. The authors used a 5-fold time series cross-validation method to obtain their performance metrics, which were root-mean-squared error (RMSE) and mean-absolute error (MAE).

For our experiments, we decided to use random samples of 63 trading days—which corresponds to a quarter of the trading days in a year—for our data sets, which can be extracted from the last 5000 trading days (from August 28th 2004 to May 18th 2020). We did not use the exactly the same data sets as Munkhdalai et al. because they do not provide the starting and ending dates for their data sets, and because a bigger data set—around 15 years of data, instead of 1 year of data—provides a better challenge for avoiding accidental "cherry picking" [44] when choosing optimal hyperparameters for the method. These data sets are split into two parts, a training data set which corresponds to 70% of the data set, and a test data set which corresponds to 30% of the data set. As a consequence, a validation step was not involved in our experiments. Regarding the performance metrics, we provide results using RMSE and MAE, in order to compare against the results presented in [1], and we also provide results using mean absolute percentage error (MAPE) and mean squared error (MSE). A total of 30 experiments were performed for each forex market, and we provide the means and standard deviations for each of our performance metrics in Section VI.

In [1], the results of several predictive models are provided. In order to obtain the hyper-parameters of the different algorithms, random searching was used, with the exception of deep learning neural networks (recurrent neural networks (RNN), long short-term memory (LSTM) neural networks, and gated recurrent unit (GRU) neural networks), where the learning rate was set at 0.001 using the Adam optimizer [45], batch size at 64 instances for each iteration, MSE as the loss function,and maximum number of epochs at 3000 for the first fold, and then they used 300 epochs with the first pre-trained model for the remaining folds. Regarding multi-layer perceptrons (MLP), the authors used an input layer of 5 neurons (for the prices of the last 5 days), a hidden layer of 16 neurons and an output layer of 1 neuron (for the prediction of the next day's price). In addition to neural network-based algorithms, [45] also provides results for models obtained by random forest, AdaBoost, XGBoost and SVM architectures.

Our MAS were optimized for 100 iterations and the loss function was set to be RMSE. The agents in the MAS have to trade at least 6 out of 9 (66%; see Section IV) trades in the same direction, regarding their activations. We obtained the parameters shown in Table 1 by trial and error. In Table 1, HH means "high height", and represents the price difference between the high and the greater price between close or open prices of a trading day; LH means "low height", and represents the price difference between the low and the lesser price between close or open prices of a trading day; CH means "candle height" and represents the absolute price

TABLE 1. Multi-agent systems parameters

Market	# of Inputs	# of Rules	Inputs
EUR/USD	9	3	HH_3, LH_3, CH_3
USD/JPY	9	6	HH_3, LH_3, CH_3
USD/CHF	12	2	HH_4, LH_4, CH_4
GBP/USD	9	3	HH_3, LH_3, CH_3
USD/CAD	12	2	HH_4, LH_4, CH_4
AUD/USD	9	3	HH_3, LH_3, CH_3

difference between the open and close prices of a trading day; and the subscript following each of the aforementioned abbreviations represents the number of past trading days that were considered.

VI. RESULTS

Tables 2 and 3 show subsets of the results presented by Munkhdalai et al. [1]. In the case of neural networks (RNN, GRU, LSTM and MLP), Tables 2 and 3 show the results obtained by Munkhdalai et al., as well as the worst and best results that are not obtained by using the activation function proposed by Munkhdalai et al. In addition to the neural network-based results, results for the predictive models based on random forest, AdaBoost, XGBoost and SVM are also provided. The purpose of these tables is to provide a comparison between the predictive models generated by our proposed method and the predictive models generated by other methods.

Our results can be found in the last rows of the tables, and the best result for each market is shown in bold. It must be noted that no statistical testing was performed when comparing our method against the ones provided by Munkhdalai et al.; the results in these tables have the purpose of showing the competitiveness of our method to some extent, in terms of error, in order to justify further research to improve our current method. Two results are presented for our method: one where activation functions (AF) are used to restrict agents from certain trades, and another for agents not using activation functions to restrict their decisions. It is noteworthy that these results cannot provide conclusive proof that one method is better than another, as the testing datasets and methods are not the same.

Table 4 provides results involving different versions of our proposed method. The table shows results for a MAS where agents have no activation functions—which means that agents respond to any set of inputs, and results for a MAS where agents have activation functions, exactly as described in Section III. The first row of the table shows the sample sizes for the first version of the method, while the second row shows the sample sizes for the second version of the method. After presenting the results for each metric, $t - values$ are presented, along with the conclusions of the hypothesis tests for each market—where we are trying to prove that the version of our method with activation functions presents lower errors than the version without activation functions. Favorable $t - values$ and conclusions are found in bold. The parameters used for all the hypothesis tests can be found in Table 5.

VII. CONCLUSION

Our method creates an architecture that successfully mixes MAS and fuzzy systems for the creation of predictive models. The generated predictive models are optimized by a simple but effective ILS.

The results, shown in Tables 2 and 3, lead us to conclude that our method proves to be competitive for predicting the six forex markets EUR/USD, USD/JPY, USD/CHF, GBP/USD, USD/CAD and AUD/USD, after being compared against models generated by RNN, LSTM, GRU, MLP, random forest, AdaBoost, XGBoost and SVM.

In the case of Table 2—which considers RMSE as the loss function—, we can see that our method performed the best without using activation functions in the agents' fuzzy systems, with the exception of GBP/USD. For EUR/USD, the method presented in [1] performed as well as our method, and outperformed ours for AUD/USD. On the other hand, Table 3—which considers MAE as the loss function—shows that our method performed the best for every forex market. An explanation of why our method performs better without activation functions when considering RMSE as the loss function is that errors are not normally distributed when the agents are being restricted by activation functions.

RMSE has been critized in the past, particularly in the works by Willmott et al. [46] [47], where RMSE is described as an "inappropriate and misinterpreted measure of average error." In these works, the authors indicate that MAE is a superior metric than RMSE for comparing predictive models.However, RMSE has also been reported to be more suitable than MAE when the error distribution is expected to be Gaussian [48], which is the case for the presented work. If we consider RMSE to be a more suitable metric to measure the performance of our models, we can conclude that our method performs better in most of the markets without the use of activation functions in the agents' fuzzy systems. Nevertheless, the use of activation functions restricts the predictive models to trade in some cases, which would yield a non-Gaussian error distribution. Considering this restriction, it may be more prudent to consider MAE as a better metric to measure the performance of our method with activation functions against our method without activation functions. After examining the hypothesis tests presented in Table 4 we can conclude that the use of activation functions dramatically improves our method's performance, if we consider MAE. However, the use of activation functions goes in detriment of the performance of our method if we consider the other metrics.

Finally, it must be noted that the results shown in Tables 2 and 3 have the purpose of demonstrating the competitiveness of our method for the creation of predictive models. Concluding that our method is better than the others shown in those tables would be naive, as the datasets used in our experiments

TABLE 2. Comparison between our results and the ones obtained by Munkhdalai et al. using RMSE as the loss function

Model	Activation function	EUR/USD	USD/JPY	USD/CHF	GBP/USD	USD/CAD	AUD/USD
RNN	Sigmoid	0.0079	6.0677	0.0097	0.0188	0.0075	0.0090
	Swish	0.0059	0.6435	0.0062	0.0099	0.0085	0.0054
	Munkhdalai et al.	**0.0057**	0.5993	0.0059	0.0098	0.0062	**0.0045**
GRU	Cosine	0.0129	2.6072	0.0168	0.0548	0.0133	0.0187
	Linear	0.0058	0.6054	0.0061	0.0104	0.0066	0.0052
	Munkhdalai et al.	0.0059	0.6871	0.0060	0.0083	0.0060	0.0082
LSTM	ReLU	0.0069	1.6890	0.0079	0.0155	0.0074	0.0058
	Swish	0.0061	0.6890	0.0072	0.0088	0.0081	0.0069
	Munkhdalai et al.	0.0062	0.6757	0.0087	0.0092	0.0078	0.0055
MLP	ReLU	0.0064	0.9711	0.0081	0.0217	0.0066	0.0048
	Swish	0.0058	0.9360	0.0061	0.0097	0.0070	0.0054
	Munkhdalai et al.	0.0061	0.7722	0.0082	0.0088	0.0064	0.0053
Random forest		0.0068	0.6781	0.0081	0.0244	0.0075	0.0056
AdaBoost		0.0076	0.8198	0.0136	0.0238	0.0080	0.0082
XGBoost		0.0076	0.6554	0.0071	0.0241	0.0085	0.0058
SVM		0.0332	0.6387	0.0394	0.0329	0.0125	0.0200
Ours without AF		**0.0057**	**0.5391**	**0.0047**	0.0079	**0.0057**	0.0052
Ours with AF		0.0071	0.5630	0.0051	**0.0078**	**0.0057**	0.0058

TABLE 3. Comparison between our results and the ones obtained by Munkhdalai et al. using MAE as the loss function

Model	Activation function	EUR/USD	USD/JPY	USD/CHF	GBP/USD	USD/CAD	AUD/USD
RNN	Sigmoid	0.0060	5.4000	0.0081	0.0126	0.0060	0.0079
	Swish	0.0044	0.4925	0.0042	0.0074	0.0068	0.0043
	Munkhdalai et al.	0.0043	0.4479	0.0037	0.0083	0.0047	0.0034
GRU	Cosine	0.0098	2.0191	0.0130	0.0396	0.0106	0.0162
	Linear	0.0044	0.4475	0.0041	0.0083	0.0052	0.0041
	Munkhdalai et al.	0.0044	0.5327	0.0039	0.0062	0.0046	0.0071
LSTM	ReLU	0.0053	1.4920	0.0054	0.0101	0.0058	0.0046
	Swish	0.0045	0.5243	0.0049	0.0065	0.0063	0.0054
	Munkhdalai et al.	0.0046	0.5200	0.0060	0.0069	0.0061	0.0044
MLP	ReLU	0.0049	0.7499	0.0058	0.0150	0.0052	0.0038
	Swish	0.0043	0.7351	0.0039	0.0073	0.0055	0.0042
	Munkhdalai et al.	0.0047	0.6114	0.0059	0.0066	0.0049	0.0041
Random forest		0.0053	0.5209	0.0061	0.0156	0.0059	0.0044
AdaBoost		0.0059	0.6440	0.0103	0.0158	0.0063	0.0066
XGBoost		0.0059	0.4958	0.0048	0.0156	0.0064	0.0045
SVM		0.0304	0.4718	0.0376	0.0294	0.0099	0.0176
Ours without AF		0.0050	0.4903	0.0041	0.0073	0.0050	0.0048
Ours with AF		**0.0033**	**0.1810**	**0.0019**	**0.0038**	**0.0033**	**0.0027**

and the methods used for testing the performance of the models differ to those used by Munkhdalai in [1]. However, it must be noted that our method has the advantage of being more interpretable than the method proposed by Munkhdalai.

VIII. FUTURE WORK

The experiments presented in this work demonstrate that better methodologies need to be used in order to provide conclusive proof that the use of activation functions in the agents' fuzzy systems is beneficial for the creation of predictive models. Additionally, we can test our activation functions in other methods, such as neural networks. Furthermore, we could design experiments that demonstrate which version of our method helps real traders make better decisions.

The proposed method was tested using a subset of all the forex markets currently available. More experiments could be performed where additional forex markets are tested. Moreover, our method should be tested with financial markets of different natures, such as stocks, bonds, commodities and metals.

An advantage of fuzzy systems over other modelling techniques is that fuzzy systems are interpretable. Natural language processing techniques that use the fuzzy systems as inputs could be used to provide texts that describe the conditions of a market, as perceived by the agents in the MAS.

TABLE 4. Results of our method with and without activation functions to restrict their actions

Metric		EUR/USD	USD/JPY	USD/CHF	GBP/USD	USD/CAD	AUD/USD
	n	60	60	60	60	60	60
	n_{AF}	50	51	56	51	50	44
MAE	Mean	5.03×10^{-03}	4.90×10^{-01}	4.13×10^{-03}	7.34×10^{-03}	5.05×10^{-03}	4.83×10^{-03}
	Std dev	2.79×10^{-03}	2.20×10^{-01}	1.94×10^{-03}	4.01×10^{-03}	2.24×10^{-03}	2.12×10^{-03}
$\mathrm{MAE_{AF}}$	Mean	3.27×10^{-03}	1.81×10^{-01}	1.95×10^{-03}	3.80×10^{-03}	3.32×10^{-03}	2.74×10^{-03}
	Std dev	3.81×10^{-03}	1.31×10^{-01}	1.53×10^{-03}	3.12×10^{-03}	1.94×10^{-03}	2.25×10^{-03}
	$t-value$	**-2.71555**	**-9.13914**	**-6.74278**	**-5.2259**	**-4.3400**	**-4.7952**
	Conclusion	**Reject H_0**	**Reject H_0**	**Reject H_0**	**Reject H_0**	**Reject H_0**	**Reject H_0**
MAPE	Mean	4.49×10^{-03}	2.89×10^{-01}	3.70×10^{-03}	6.53×10^{-03}	4.51×10^{-03}	4.32×10^{-03}
	Std dev	2.47×10^{-03}	1.01×10^{-01}	1.73×10^{-03}	3.50×10^{-03}	1.99×10^{-03}	1.88×10^{-03}
$\mathrm{MAPE_{AF}}$	Mean	6.04×10^{-03}	2.93×10^{-01}	3.92×10^{-03}	6.10×10^{-03}	4.53×10^{-03}	5.53×10^{-03}
	Std dev	4.38×10^{-03}	1.50×10^{-01}	2.47×10^{-03}	3.19×10^{-03}	2.07×10^{-03}	3.20×10^{-03}
	$t-value$	2.2248	0.1618	0.5520	-0.6768	0.0513	2.2406
	Conclusion	Fail to reject H_0	Fail to reject H_0	Fail to reject H_0	Fail to reject H_0	Fail to reject H_0	Fail to reject H_0
MSE	Mean	4.20×10^{-05}	3.52×10^{-01}	2.76×10^{-05}	9.21×10^{-05}	3.92×10^{-05}	3.35×10^{-05}
	Std dev	5.01×10^{-05}	3.37×10^{-01}	2.98×10^{-05}	1.36×10^{-05}	4.06×10^{-03}	3.12×10^{-05}
$\mathrm{MSE_{AF}}$	Mean	6.81×10^{-05}	2.62×10^{-01}	2.95×10^{-05}	6.46×10^{-05}	3.62×10^{-05}	5.25×10^{-05}
	Std dev	9.54×10^{-05}	2.52×10^{-01}	3.55×10^{-05}	6.77×10^{-05}	3.29×10^{-05}	5.49×10^{-05}
	$t-value$	1.7444	-1.6066	0.3111	**-2.8524**	-0.0057	2.0642
	Conclusion	Fail to reject H_0	Fail to reject H_0	Fail to reject H_0	**Reject H_0**	Fail to reject H_0	Fail to reject H_0
RMSE	Mean	5.69×10^{-03}	5.39×10^{-01}	4.69×10^{-03}	7.93×10^{-03}	5.70×10^{-03}	5.18×10^{-03}
	Std dev	3.18×10^{-03}	2.51×10^{-01}	2.09×10^{-03}	4.83×10^{-03}	2.64×10^{-03}	2.34×10^{-03}
$\mathrm{RMSE_{AF}}$	Mean	7.12×10^{-03}	5.63×10^{-01}	5.11×10^{-03}	7.79×10^{-03}	5.70×10^{-03}	5.81×10^{-03}
	Std dev	5.15×10^{-03}	2.98×10^{-01}	3.21×10^{-03}	4.05×10^{-03}	2.36×10^{-03}	3.72×10^{-03}
	$t-value$	1.7104	0.4543	0.8288	-0.1661	0.0000	0.9890
	Conclusion	Fail to reject H_0	Fail to reject H_0	Fail to reject H_0	Fail to reject H_0	Fail to reject H_0	Fail to reject H_0

TABLE 5. Parameters used for the hypothesis tests

Parameter	Value
Confidence interval	95%
H_a	$\mu_1 < \mu_2$
H_0	$\mu_1 \geq \mu_2$
Critical t	-1.9958

Our implementation should be adapted to perform in a distributed manner, where agents are evaluated in parallel in different CPU cores or even different physical machines. A distributed architecture will help achieve results faster, which will allow us to test different approaches for the creation of predictive models using our method.

Finally, the ILS method used for optimization in the proposed method should be tested in performance against other optimization algorithms, such as genetic algorithms or particle swarm optimization. Finding better optimization algorithms for our method can help us achieve results faster. It is also probable that lower errors could be achieved, as we do not know if our current optimization method is considering an appropriate search space.

ACKNOWLEDGMENT

This paper has been supported in part by project DeepBio (TIN2017-85727-C4-2-P).

AMAURY HERNANDEZ-AGUILA

REFERENCES

[1] L. Munkhdalai, T. Munkhdalai, K. H. Park, H. G. Lee, M. Li, and K. H. Ryu, "Mixture of Activation Functions with Extended Min-Max Normalization for Forex Market Prediction," IEEE Access, vol. 7, pp. 183 680–183 691, 2019.

[2] G. Liu and X. Wang, "A Numerical-Based Attention Method for Stock Market Prediction with Dual Information," IEEE Access, vol. 7, no. c, pp. 7357–7367, 2019.

[3] Y. Alsubaie, K. E. Hindi, and H. Alsalman, "Cost-Sensitive Prediction of Stock Price Direction: Selection of Technical Indicators," IEEE Access, vol. 7, pp. 146 876–146 892, 2019.

[4] D. Lien Minh, A. Sadeghi-Niaraki, H. D. Huy, K. Min, and H. Moon, "Deep Learning Approach for Short-Term Stock Trends Prediction Based on Two-Stream Gated Recurrent Unit Network," IEEE Access, vol. 6, no. c, pp. 55 392–55 404, 2018.

[5] D. Cabrera, C. Cubillos, A. Cubillos, E. Urra, and R. Mellado, "Affective Algorithm for Controlling Emotional Fluctuation of Artificial Investors in Stock Markets," IEEE Access, vol. 6, no. c, pp. 7610–7624, 2018.

[6] S. M. Idrees, M. A. Alam, and P. Agarwal, "A Prediction Approach for Stock Market Volatility Based on Time Series Data," IEEE Access, vol. 7, no. c, pp. 17 287–17 298, 2019.

[7] W. Cao, W. Zhu, and Y. Demazeau, "Multi-Layer Coupled Hidden Markov Model for Cross-Market Behavior Analysis and Trend Forecasting," IEEE Access, vol. 7, pp. 158 563–158 574, 2019.

[8] Y. Guo, S. Han, C. Shen, Y. Li, X. Yin, and Y. Bai, "An Adaptive SVR for High-Frequency Stock Price Forecasting," IEEE Access, vol. 6, no. c, pp. 11 397–11 404, 2018.

[9] Y. Chen, W. Lin, and J. Z. Wang, "A Dual-Attention-Based Stock Price Trend Prediction Model with Dual Features," IEEE Access, vol. 7, pp. 148 047–148 058, 2019.

[10] J. A. Jiang, C. H. Syue, C. H. Wang, J. C. Wang, and J. S. Shieh, "An Interval Type-2 Fuzzy Logic System for Stock Index Forecasting Based on Fuzzy Time Series and a Fuzzy Logical Relationship Map," IEEE Access, vol. 6, no. c, pp. 69 107–69 119, 2018.

[11] X. Sang, Y. Zhou, and X. Yu, "An Uncertain Possibility-Probability Information Fusion Method Under Interval Type-2 Fuzzy Environment and Its Application in Stock Selection," Information Sciences, vol. 504, pp. 546–560, 2019. [Online]. Available: https://doi.org/10.1016/j.ins.2019.07.032

[12] K. Chourmouziadis and P. D. Chatzoglou, "Intelligent Stock Portfolio Management Using a Long-Term Fuzzy System," Applied Artificial Intelligence, vol. 33, no. 9, pp. 775–795, 2019. [Online]. Available: https://doi.org/10.1080/08839514.2019.1630124

[13] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," International Journal of Man-Machine Studies, vol. 7, no. 1, pp. 1–13, 1975.

[14] M. Abdulgader and D. Kaur, "Evolving Mamdani Fuzzy Rules Using Swarm Algorithms for Accurate Data Classification," IEEE Access, vol. 7, pp. 175 907–175 916, 2019.

[15] K. T. Atanassov, "Intuitionistic fuzzy sets," pp. 87–96, 1986. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165011486800343

[16] ——, "Intuitionistic Fuzzy Sets Past, Present and Future," in EUSFLAT Conf., vol. 1, 2003, pp. 145–149. [Online]. Available: http://ifigenia.org/mediawiki/images/d/d3/EUSFLAT-2003-012-019.pdf

[17] A. Hernandez-Aguila, M. Garcia-Valdez, and O. Castillo, "A proposal for an intuitionistic fuzzy inference system," 2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 1294–1300, 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7737838/

[18] M.-C. Tsai, C.-H. Cheng, and M.-I. Tsai, "A Multifactor Fuzzy Time-Series Fitting Model for Forecasting the Stock Index," Symmetry, vol. 11, no. 12, p. 1474, 2019.

[19] S. Zeng, S. M. Chen, and M. O. Teng, "Fuzzy Forecasting Based on Linear Combinations of Independent Variables, Subtractive Clustering Algorithm and Artificial Bee Colony Algorithm," Information Sciences, vol. 484, pp. 350–366, 2019. [Online]. Available: https://doi.org/10.1016/j.ins.2019.01.071

[20] S. Rajab and V. Sharma, "An Interpretable Neuro-Fuzzy Approach to Stock Price Forecasting," Soft Computing, vol. 23, no. 3, pp. 921–936, 2019.

[21] A. Vlasenko, N. Vlasenko, O. Vynokurova, Y. Bodyanskiy, and D. Peleshko, "A Novel Ensemble Neuro-Fuzzy Model for Financial Time Series Forecasting," Econometrics in Theory and Practice, pp. 439–453, 2019.

[22] W. Yue, Y. Wang, and H. Xuan, "Fuzzy Multi-Objective Portfolio Model Based on Semi-Variance–Semi-Absolute Deviation Risk Measures," Soft Computing, vol. 23, no. 17, pp. 8159–8179, 2019. [Online]. Available: https://doi.org/10.1007/s00500-018-3452-y

[23] W. Witayakiattilerd, "Fuzzy Quantitative Analysis Method for Stock Selection Into Portfolio," Chiang Mai Journal of Science, vol. 46, no. 4, pp. 799–811, 2019.

[24] N. Mansour, M. S. Cherif, and W. Abdelfattah, "Multi-objective Imprecise Programming for Financial Portfolio Selection With Fuzzy Returns," Expert Systems with Applications, vol. 138, p. 112810, 2019. [Online]. Available: https://doi.org/10.1016/j.eswa.2019.07.027

[25] B. Lebaron, "A Builder's Guide to Agent Based Financial Markets," Quantitative finance, vol. 1, pp. 254–261, 2001.

[26] R. Grothmann, "Multi-Agent Market Modeling Based on Neural Networks," Faculty of Economics, University of Bremen, 2002. [Online]. Available: http://lsc.fie.umich.mx/ juan/Materias/Cursos/ANN/Papers/multi-agent-market-modeling.pdf

[27] Y. Li, C. Wu, J. Liu, and P. Luo, "A Combination Prediction Model of Stock Composite Index Based on Artificial Intelligent Methods and Multi-Agent Simulation," International Journal of Computational Intelligence Systems, vol. 7, no. 5, pp. 853–864, 2014.

[28] X. Chen, "Multi-Agent-Based Modeling of Artificial Stock Markets by Using the Co-Evolutionary GP Approach," Journal of the Operations Research Society of Japan, 2004.

[29] J. J. Kim, S. H. Cha, K. H. Cho, and M. Ryu, "Deep Reinforcement Learning Based Multi-Agent Collaborated Network for Distributed Stock Trading," International Journal of Grid and Distributed Computing, vol. 11, no. 2, pp. 11–20, 2018.

[30] L. Wei, W. Zhang, X. Xiong, and Y. Zhao, "A Multi-Agent System for Policy Design of Tick Size in Stock Index Futures Markets," Systems Research and Behavioral Science, vol. 31, no. 4, pp. 512–526, 2014.

[31] M. Gao, S. Yang, and L. Sheng, "Distributed Fault Estimation for Time-Varying Multi-Agent Systems with Sensor Faults and Partially Decoupled Disturbances," IEEE Access, vol. 7, pp. 147 905–147 913, 2019.

[32] C. Yu, X. Chen, C. Wang, H. Wu, J. Sun, Y. Li, and X. Zhang, "An Improved Platform for Multi-Agent Based Stock Market Simulation in Distributed Environment," IEICE Transactions on Information and Systems, vol. E98D, no. 10, pp. 1727–1735, 2015.

[33] L. A. Zadeh, "Fuzzy sets," Information and Control, no. 8, pp. 338–3365, 1965. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S001999586590241X

[34] R. Kruse, J. E. Gebhardt, and F. Klowon, Foundations of fuzzy systems. John Wiley & Sons, Inc., 1994.

[35] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," IEEE transactions on systems, man, and, 1985.

[36] J. M. Mendel and R. I. John, "Type-2 Fuzzy Sets Made Simple," IEEE Transactions on fuzzy systems, 2002. [Online]. Available: http://ieeexplore.ieee.org/abstract/document/995115/

[37] J. M. Mendel, R. I. John, and F. Liu, "Interval Type-2 Fuzzy Logic Systems Made Simple," IEEE transactions on fuzzy systems, 2006. [Online]. Available: http://ieeexplore.ieee.org/abstract/document/4016089/

[38] A. Hernandez-Aguila, M. Garcia-Valdez, O. Castillo, and J. J. M. Guervós, "An open source implementation of an intuitionistic fuzzy inference system in Clojure," IEEE International Conference on Fuzzy Systems, pp. 1–6, 2017.

[39] S. Zhao, E. C. Tsang, D. Chen, and X. Wang, "Building a rule-based classifier—a fuzzy-rough set approach," IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 5, pp. 624–638, 2009.

[40] M. Almutairi, F. Stahl, and M. Bramer, "A rule-based classifier with accurate and fast rule term induction for continuous attributes," in 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE, 2018, pp. 413–420.

[41] M. Wygralak, "An axiomatic approach to scalar cardinalities of fuzzy sets," Fuzzy sets and Systems, vol. 110, no. 2, pp. 175–179, 2000.

[42] Y. Shoham, "Agent-oriented programming," Artificial Intelligence, vol. 60, no. 1, pp. 51–92, 1993.

[43] M. Johnson, "Memoization of top down parsing," arXiv preprint cmp-lg/9504016, 1995.

[44] J. M. Morse, ""cherry picking": Writing from thin data," 2010.

[45] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.

[46] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance," Climate research, vol. 30, no. 1, pp. 79–82, 2005.
[47] C. J. Willmott, K. Matsuura, and S. M. Robeson, "Ambiguities inherent in sums-of-squares-based error statistics," Atmospheric Environment, vol. 43, no. 3, pp. 749–752, 2009.
[48] T. Chai and R. R. Draxler, "Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature," Geoscientific model development, vol. 7, no. 3, pp. 1247–1250, 2014.